System Services CSCI

Access Control and Security CSC

Thor Design Panel 2/3 Review

84K00510-030

December 16, 1997

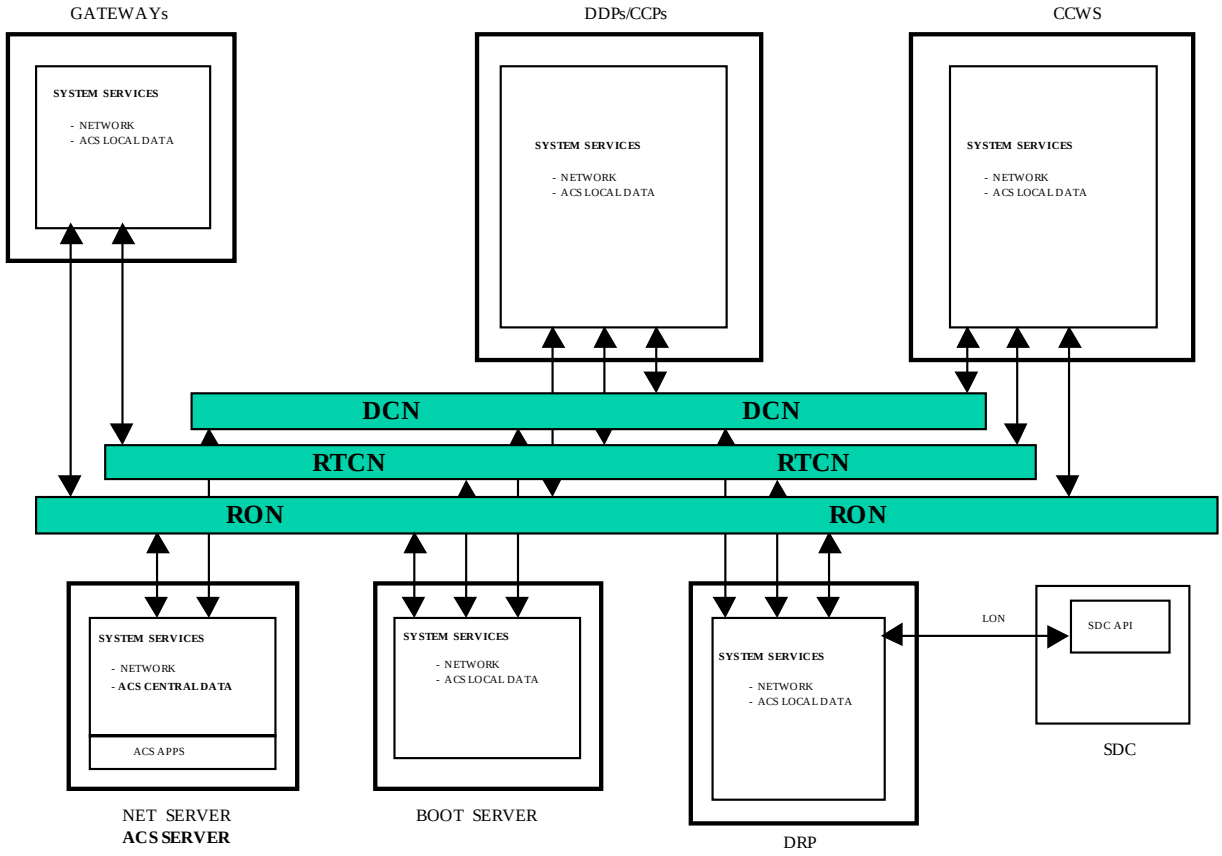
Tom Nguyen

1. Access Control and Security

1.1 Access Control and Security Introduction

1.1.1 Access Control and Security Overview

The Access Control and Security (ACS) CSC provides access control and security policies for RTPS. ACS will be implemented over a series of CLCS deliveries and THOR will cover a minimal set of requirements. These policies will address user and system access, system file integrity, system security, and system auditing requirements. The boundaries of the ACS measurement are the network connection point of the individual system and it's internal configuration (i.e. Network traffic/data is not considered as part of this CSC). The ACS CSC are the software components that applies to each machine, and provides control and auditing capabilities using standard built-in COTS OS procedures and features.



SYSTEM SERVICES OVERVIEW

1.1.2 Access Control and Security Description

Access Control and Security (ACS) can be divided into server and clients. The server is a secure (restricted access) machine used for monitoring, logging, and auditing of all client systems. The server maintains the central ACS data. The clients will trust and send all ACS data to the server. The server will periodically transfer the central ACS data to tape for storage and future retrieval if necessary. In the operational environment this storage will be a local tape drive with a minimum capacity as specified in section 1.2.1. However, in the development environment this storage may be via the existing Auspex File Server (or designated equivalent) where ACS data will reside and backup with normal daily system backup functions maintained by the development team (i.e. System Services - OS group). ACS provides a method of configuring, archiving, and retrieving ACS data for the various platforms and computing bases in RTPS.

1.2 Access Control and Security Specifications

1.2.1 Access Control and Security Groundrules

The following is a list of groundrules and assumptions that relate to the Access Control and Security CSC for THOR:

- The scope of ACS will be limited to RTPS.
- Security policies and guidelines are determined by Security System Engineering.
- Firewall and network security including network monitoring, filtering, scanning, etc. is currently being implemented and will be re-evaluated in future deliveries for future requirements.
- ACS implementation will be provided by available built-in COTS software.
- ACS for Gateways will not be implemented for THOR (TBD for future releases due to vendor OS).
- Access Control and Security server shall have access to all RTPS platforms.
- ACS data will be in tape (4MM DAT) format.
- The central ACS data is not expected to exceed a maximum of 1GB of data per month to be transferred to tape (assuming a maximum of 50 hosts).

1.2.2 Access Control and Security Functional Requirements

1.2.2.1. ACS will be implemented on the following platforms:

a. CCP
b. DDP
c. CCWS
d. Boot Server
e. Net Server (ACS Server)
f. *Gateways (TBD)*
g. *DRP (TBD)*

1.2.2.1. All local ACS data from listed clients will be collected and archived to a central server.
1.2.2.2. ACS will provide the capability to retrieve archived ACS data using available built-in COTS software.
1.2.2.3. ACS will provide a method and procedures for tape backups to facilitate ACS retrieval.
1.2.2.4. All clients will have a trust relationship to the ACS server for access control recording.
1.2.2.5. All clients and servers will have a security banner prescribed by NASA for government computers when a login prompt is presented.
1.2.2.6. ACS will record all client user and system access (login) and attempts.
1.2.2.7. ACS clients will log security related warning and critical messages for services such as:

a. telnet
b. ftp
c. rsh
d. rcp
e. rlogin
f. xhost

1.2.2.1. ACS will record all Super User (privileged user) accesses on all clients and servers.
1.2.2.2. ACS will disable unnecessary accounts (i.e. demo, games, nobody, etc.) on all clients and servers.
1.2.2.3. ACS will disable all unnecessary network services (i.e. sendmail, httpd, tftpd, finger, programs/daemons) on all clients and servers.
1.2.2.4. ACS will verify shared data (exported filesystems) for access security (i.e. writable by everyone) on all clients and servers.

1.2.3 Access Control and Security Performance Requirements

No specific requirements for THOR.

1.3 Access Control and Security Design Specification

The Access Control and Security design concept is generic to vendor platforms. The mechanisms and procedures to accomplish ACS configuration are specific to a vendor. The steps of implementing the Access Control and Security on server and clients are described in section 1.2.2. This client and server configuration and tunable parameters are facilities built-in to COTS UNIX OS. The facilities are described in details in Appendix A.

The ACS configuration for client and server includes but is not limited to the following steps and may have vendor specific syntax other than what is shown:

1) Configure /.rhosts and/or /etc/hosts.equiv file to include ACS server.
2) Configure /etc/inetd.conf to disable unecessary services.
3) Configure /etc/exports with proper permissions.
4) Edit /etc/passwd, /etc/group, and/or NIS/YP files to disable unecessary accounts.
5) Check and disable other unnecessary services which are vendor specific (i.e. on SGI use chkconfig).
6) Check and disable routing and IP forwarding unless is required for the platform.
7) Configure /etc/syslog.conf to log to the ACS server (on for clients).

The ACS server configuration to record the data from the client machines also includes applying appropriate filters for ACS specific data such as user and system access, system file integrity, and system security. This recorded data is considered as the central ACS data. Once a month the ACS server shall archive to tape media. For THOR, a 4MM DAT tape will be the media.

Should retrieval be needed from tape, the appropriate time and date stamped tape and ACS retrieval tool can be used to display retrieved data. The retrieval tools consisting of scripts which restores the data from tape and then displays it on screen with built-in COTS software such as *system log viewer* where data can be filtered by date, priority, type, etc.

1.3.1 Access Control and Security Detailed Data Flow

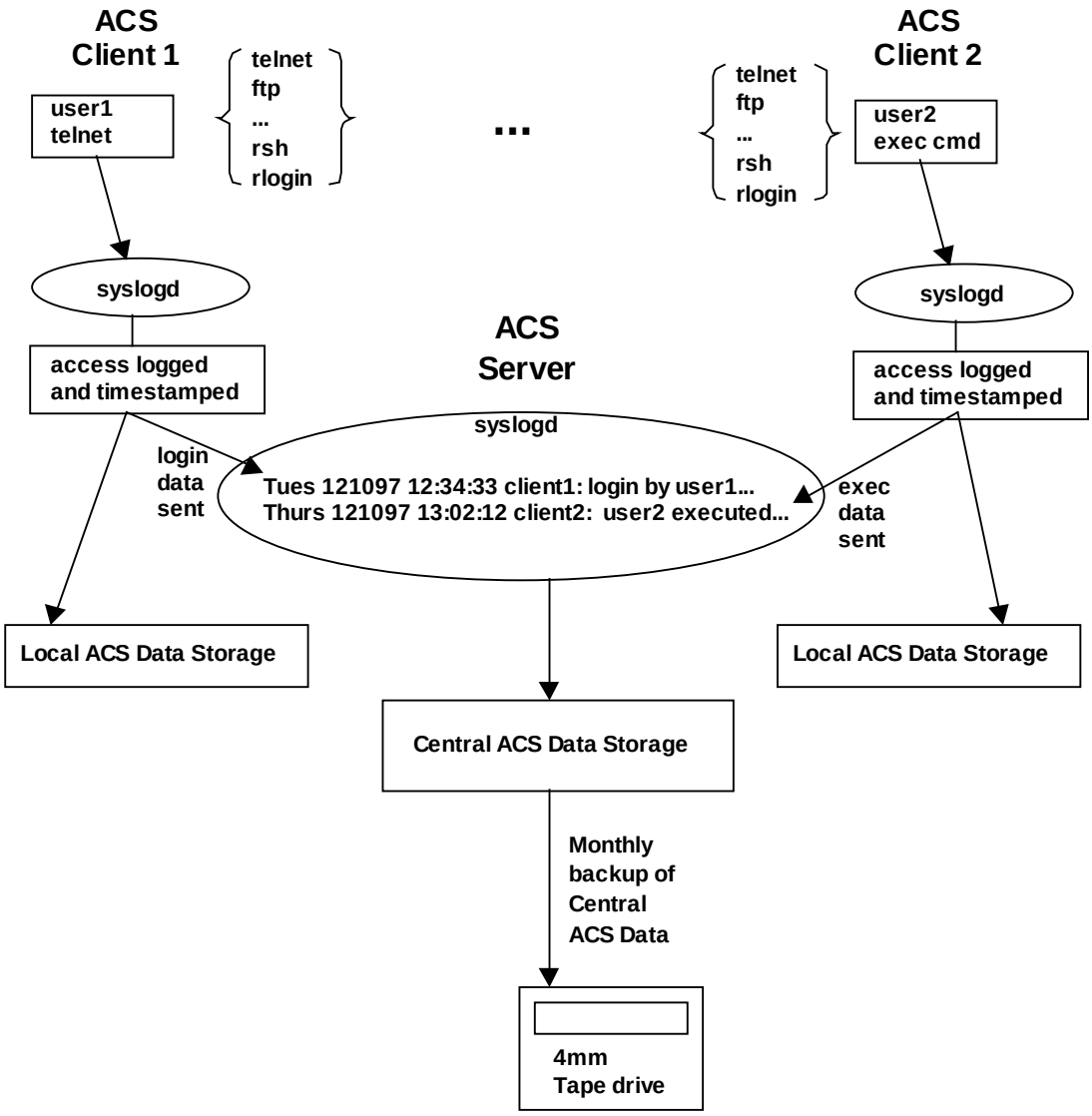
This data flow provides a pictorial representation of the flow between an ACS client and an ACS server upon a user using a system service.

In the examples shown, user1 accesses the system via, telnet, a system service on ACS client1. A built-in UNIX facility, syslogd, reads the kernel message and records the access with appropriate date, time, priority code, facility code, client hostname, service name (login), process ID, and status or result (user ID succeeded or failed). The ACS client1 immediately sends this data to the ACS server as configured in /etc/syslogd.conf.

The ACS server receives the message from client1 through its own syslogd, passes the data through a filter process (program which parses out appropriate ACS security data), and records the data to the server's disk (Central ACS Data storage). This cyclic process of data capturing, filtering, and recording occurs for all messages received by the ACS server.

In similar actions, user2 uses rsh to execute a command on ACS client2. The ACS client2 receives and processes the kernel message via syslogd and passes the message to the ACS server. The server then performs the same cyclic process.

ACS Data Flow Process



For ACS data archiving, once per month using a UNIX available function (i.e. cronjob), the ACS data is saved to tape media using a UNIX utility (i.e. tar). For ACS data retrieval, the data will be restored from the appropriate tape archive using scripts which recover the data (also using a UNIX tar utility) into a local directory and launches an application to view the ACS data. In the vendor specific platform such as SGI, the system log viewer (shown in section 1.3.2.2) is launched and the data can be filtered and viewed.

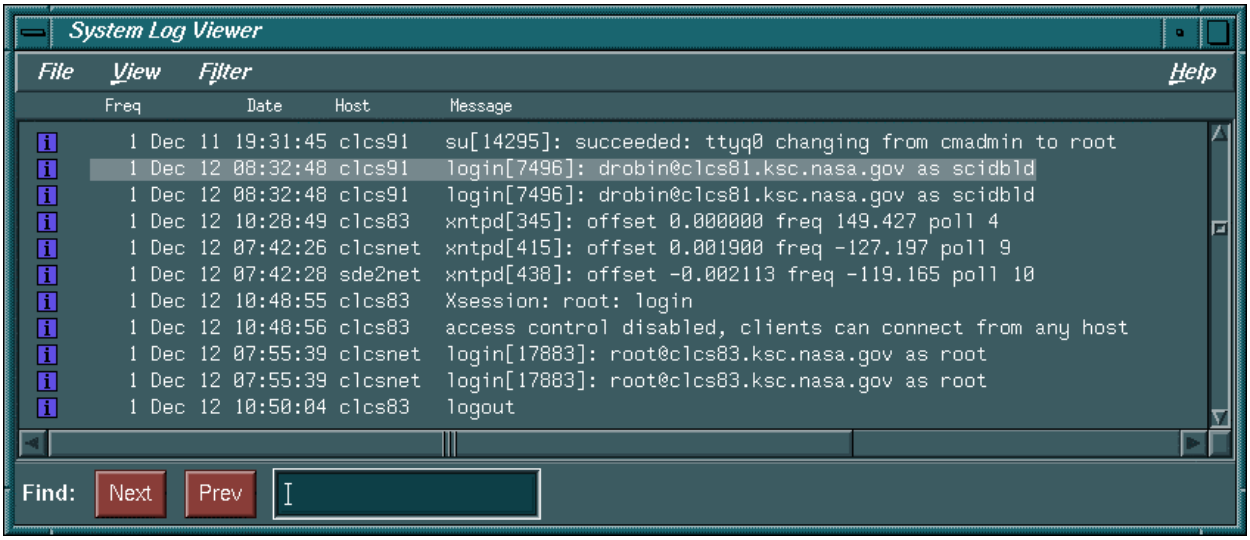
1.3.2 Access Control and Security CSC External Interfaces

1.3.2.1 Access Control and Security Message Formats

PRIORITY CODE	FACILITY CODE	DATE	TIME	HOSTNAME	SERVICE	PROCESS ID	STATUS
6	E	Dec 12	13:12:34	Ide1hci2	login	23421	Failed: root@ide1hci.ksc.nasa.gov as root
6	E	Dec 12	13:14:34	Ide1hci2	login	23622	root@ide1hci.ksc.nasa.gov as root
6	E	Dec 12	13:16:34	Ide1hci2	rshd	23733	Failed: root@ide1hci.ksc.nasa.gov as root cmd='grep ide1 /etc/hosts'
6	E	Dec 12	13:18:34	Ide1hci2	rshd	23946	root@ide1hci.ksc.nasa.gov as root cmd='grep ide1 /etc/hosts'

Priority code - Listed as 0-7 (EMERG, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFO, DEBUG).
Facility code - Listed as A-T (ranging from kernel messages to network to user messages see UNIX man pages for syslogd and see /usr/include/sys/syslog.h).
NOTE: Priority and facility codes are encoded into a single 32-bit quantity, where the bottom 3 bits are the priority and the top 28 bits are the facility (0-big number). Both codes are map roughly one-to-one to strings in the syslogd source code.
Date – Format as Month abbreviated and day.
Time – 24 hr format.
Hostname – hostname of system.
Service – system service or program name.
Process ID – integer numbers assigned to each service or program when executed in UNIX OS.
Status – Result of the execution of service (i.e. failure, success, comands, actions).

1.3.2.2 Access Control and Security Display Formats



1.3.2.3 Access Control and Security Input Formats

NA

1.3.2.4 Access Control and Security Printer Formats

NA

1.3.3 Access Control and Security Test Plan

ACS Validation and Platform Operations testing will consist of procedures and utilities to test key components and aspects of each platform and function. The key areas include but are not limited to:

1. Configuration of ACS client and server
2. ACS client logging to ACS server
3. Archiving and retrieving of Central ACS data

1.3.3.1 Test Environment

ACS Validation and Platform testing will consist of processes common to all environments and localized tests, which are unique to each specific environment. Common tests will be ones such as Software Comparator Tests to address areas which are intended to be common throughout the environment (IDE1, LCCX, SDE1, SDE2, CLCS, etc). Localized tests will address those areas which are necessarily unique (network addressing, server connectivity, etc).

1.3.3.2 Test Methods

Testing methods will be constructed to perform in a non-intrusive manner. They will use utilities and commands readily available on each platform and be designed to minimize customization. Unix command-line testing will include standard Unix commands such as **diff, cat, ls, ping, ifconfig, grep**, and others.

1.3.3.3 Test Cases

1.3.3.4. Configuration of clients and server

Testing of client functionality includes verification that the clients are configured and ACS data are collected and sent to the central ACS server. This is not limited to only following the steps described in earlier section 1.3 but to utilize available UNIX commands to verify functionality.

1.3.3.5. ACS client logging to ACS server

Perform test on services listed in section 1.2.2.8 on different client platform type listed in section 1.2.2.1. Verification by inspection that the ACS client data is contained in the ACS server logs and the format of logs are properly stored. Verify by inspection that each ACS client platform type is properly recorded on ACS server.

1.3.3.6. Archiving and retrieving of Central ACS data

Verification of ACS data are properly stored on tape media on a monthly interval and data is intact. This can be done by (1) examining ACS server configuration for UNIX crontab for commands to archive ACS data and (2) restoring from tape the ACS data and verified by inspection that data is in original saved format.

Appendix A

Extracted from UNIX man pages:

```
syslogd - log systems messages

syslogd reads and logs messages into a set of files described by the
configuration file /etc/syslog.conf.  Each message is one line.  A message can
contain a priority code, marked by a number in  angle braces at the beginning
of the line.  Priorities are defined in <sys/syslog.h>.

syslogd reads from the stream device /dev/log, from an Internet domain socket
specified in /etc/services, and from the special device /dev/klog (to read
kernel messages).

Lines in the configuration file have a selector to determine the message
priorities to which the line applies and an action.
For example, the configuration file:

        kern.debug  |/usr/sbin/klogpp            /var/adm/SYSLOG
        kern.debug  |/usr/sbin/klogpp            /dev/console
        user,mail,daemon,auth,syslog,lpr.debug /var/adm/SYSLOG
        kern.err    @ginger
        *.emerg     *
        *.alert     eric,beth
        *.alert;auth.warning              ralph

filters all kernel messages through klogpp(1M) and writes them to the
system console and into /var/adm/SYSLOG and logs debug (or  higher)level
messages into the file /var/adm/SYSLOG.  Kernel messages of error everity or
higher are forwarded to ginger.  All users are  informed of any emergency
messages.  The users eric and beth are informed of any alert messages.  The
user ralph is informed of any alert message or any warning message (or higher)
from the authorization system.


     This example shows how to use the filter mechanism.  To have ftpd(1M)
     messages logged in a different file, add the following line to
     /etc/syslog.conf:

        daemon,auth.debug       |/var/adm/ftpd.filt  /var/adm/ftpd.log

     The /var/adm/ftpd.filt file is a shell script:

        #!/bin/sh
        # This filter only accepts ftpd messages
        read line
        set $line
        case "$1" {
            ftpd\[*)
              echo "$line\c"
              exit 0
              ;;
        }
```

```
        exit 0
```

The following is an example line from the /var/adm/SYSLOG file:

```
   Aug 10 10:32:53 6F:sgihost syslogd: restart
```

Each line has several parts. The date and time of the message are listed first, followed by a priority and facility code. Priorities are listed as 0-7 and facilities are listed as A-T.

Reference <sys/syslog.h>. The source is the name of the program that generated the message. Following the source is the message itself.

```
/etc/syslog.conf            default configuration file
/dev/log                    device read by syslogd
/dev/klog                   the kernel log device
/usr/sbin/klogpp            filter for kernel messages
/etc/config/syslogd.options   command-line flags used at system startup
```